				Τ	<u> </u>	ſ	
			102		100		Ħ
							IVR APP
			Init VSR by invoking Init() method that VSR Interface provides		Loaded and init itself then Create Instance of VSR (DAL)		PBX
		"KeepAlive", Heart Beat with 1 second interval, packet to all PEUs using Control packet queue.	VSR start broadcasting	VSR loaded			VSR (DAL)
							PEU Object
							WAV Port
							Actual Port
							WAV Driver
	DD passes packet to FPGA					Loaded at the time of system up	Device Driver & FPGA
Upon receipt of "Keep Alive" packet, sends "Hello" packet to VSR indicating ID and other information such as number of ports, SN. For all PEUs							PEU and DSP

UULFAU WEGWAGE

E		- PBX	VSR (DAL)	\	WAV	Actual	*	Device Driver & FPGA
	APP			Object Port	Port	Port	Driver	
					·			
	i							DD passes packet to
								Shared memory
			VSR creates PEU Interface					
			Instance for each Physical					
			PEU detected.					
			Based on reported information,					
			PEU Interface Instance					
			provides table for ports.				:	
				Created				

	Ħ
	IVR APP
PBX enumerates all PEUS and gets information of PEU through PEU Interface Instances those created to drive Physical PEUs. As PBX finds a matched Actual Port from the configuration, PBX invokes PEU Interface method to create and Actual Port Object for each specified port number.	PBX
	VSR (DAL)
PEU Interface Instance looks up specified port number in table and invokes Create Port Interface Instance as needed to create an Actual Port Object for each actual port specified by the PBX. Return handle of created Port Interface Instance (Actual Port Object) to	PEU Object
	Port
	Actual Port
	WAV Driver
FPGA	
	and DSP

ا ٦				•	1					\top				T					-	_
3	E																			_
1177 A A A A	IVK APP			IVR App loaded by operator and attaches WAV Driver to	its process area.					A -1- DDV 6 AWAW	Ask PBX for a WAV	Channel through TAPI	based on its configuration	table						
7,747	PBX	PBX finds all handles for	its ports configured in the table.												PBX looks up the next	WAY shannel in its table	Invoke VSR Interface	method to create WAV	Port with specified Device	3
	(DAL)																			
	Object	_																		
	Port			,													-			
	Actual Port									1										
	WAV Driver					and loaded.	Number of	devices defined at	the time of	compilation.										
	Device Driver & FPGA												-							
	and DSP												-					4		

uurrau saaamau

						- T	_			
IVK APP						IVR App keeps Device	ID for future use.	It repeats for the number	of WAV ports in its	configuration
PBX				PBX provides	application	through TAP1				
VSR (DAL) VSR looks up WAV channel ID in its table and, if no WAV Port already created with that device ID, creates WAV Port Interface Instance (WAV port) and sets Stream ID to Device ID VSR returns handle of		VSR returns handle of	created WAV Port							
Object										
WAV Port Created	Created									
Port									-	
Driver							- 12	•		
Device Driver & FPGA								•	•••	
DSP								<u> </u>		4

OUTTO SESTE

Run Time

B		<u> </u>		112								110					108				Š	12	104		Ш
IVR APP								·			_													APP	IVR
PP																									PBX
РВХ					was created for port that detected CO Ring.	VSR dispatches message to the Actual Port Object that	pointers	device driver and receiving	generates a software	in response to an event that	the shared memory either	VSR reads packets from													VSR (DAL)
					ort that	nessage to bject that		l receiving	/are	event that	ry either	ts from								,					
VSR										•											-			Object	PEU
VSR (DAL)																								Port	WAV
PEU Obje	Interface	PBX via COM	the RING	Actual Port																					Actual Port
ct		OM	b 303	t can						-															rt
WAV Port																								Driver	WAV
Actual Port													into shared memory	DAL by writing	queue and copies to	packet in FPGA	Device Driver sees							FPGA	Device Driver &
WAV Driver													memory	riting	copies to	PGA	ver sees								iver &
Device Driver &																		driver	sends nacket	PETI and Port info then	generates nacket with	MC Datasta	Specific nort		PEU and DSP
PEU and																			to device	t info then	status,	Ctatus	lg on one		SP

		,	, , ,						
ID	128	126		124	Ш	122	116	114	
IVR APP					IVR APP		IVR Inv function PBX to call		
РВХ				Object n port to c function	PBX		IVR Invokes TAPI function call to order PBX to answer the call		
VSR (DAL)				mapped do "Go			er		
AL)				PBX orders Actual Port Object mapped to ringing port to do "Go Offhook" function		PBX receives order from the TAPI App to answer the call		PBX invokes "New Call" function to IVR through TAPI	
PEU Object					VSR (DAL)	es order PI App le call		s "New on to IVR PI	
WAV					PEU Object				
Actual Port					WAV Port				
			mapped port in "Go O with de and Po Then whered	Achial	Actual Port				
WAV Driver			mapped to ringing port in VSR generates "Go Offhook" packet with destination PEU and Port address. Then write it into shared memory.	Actual Port Object	Port				
Device Driver &			ing ing nerates packet n PEU ss.	2					
80					WAV Driver				
PEU and DSP	packet via pac or othe	Device control FIFO n			Device				
)SP	transmit to addre ket swit r contro	Device Driver sends control packet to FP FIFO memory.			Driver				4
	FPGA transmits control packet to addressed PEU via packet switched bus or other control channel	Device Driver sends control packet to FPGA FIFO memory.			Device Driver & FPGA				FPGA
					PEU and DSP				DSP
1	i	l i		1	1			ı l	i 1

								138				136			134	i		132				130	
										-													
		C	. .			· · · ·																	
		Object	Dispatch message to addressed														٠						
									:														
			.,	:																			
status change message to PBX	Actual Port Object sends																						
																		18					
			-	DAL DAL	,															.,,			FPGA
					DAL.	packet through Device Driver and	PBX by MC by upstream control	Loop Current status is reported to	status.	And the MC detects this change in	interface circuit detects loop current.	A Loop Current detector in the line	cause it to "Go Offhook".	SLIC for port with ringing CO line to	MC writes control bit to appropriate	or by polling.	FIFO in PEU in response to interrupt	MC sweeps out received packets from	them in FIFO	packets addressed to the PEU and puts	then selects broadcast packets or	FPGA in PEU scans PEU address	

UUTTIN SKESHEG

		144		140	Ш
			IVR App invoke TAPI function requesting PBX to switch WAV Port of specified Device ID to couple it to the Actual Port Object mapped to the port that just went to "Offhook"		IVR APP
		PBX invokes function of WAV Port Interface instance specified by the Device ID to connect to the actual port to listen.		PBX reports Port Status to IVR App through TAPI.	РВХ
					VSR (DAL)
					PEU Object
	Wav port with device ID specified by IVR app registers as listener to specified actual port.				WAV Port
WAV Port registered.					Actual Port
				,	WAV Driver
					Device Driver & FPGA
					PEU and DSP

UULTIU BAĞBISGO

					_				•	<u> </u>											,		146		E	•
			•																						APP	1477
																				WAV Port to listen.	instance to connect to	Port Interface	PBX invokes		PBX	
																								(22.27)	DAI:	175
			-																					1	Ohiec	77.7
								ID of the WAV Port	Stream ID the Device	data using as a	start playing WAV	Actual Port Object to	function call of	WAV Port invokes	Actual Port registered										WAV Port	
		memory	writes it to shared	Stream ID	Play" packet with	Object responds by generating a "Start	The Actual Port									Port	to specified WAV	registers as listener	Actual Port Object						Actual Port	
																			ń						WAV	4.
packet to FPGA and FPGA delivers it to the PEU	Pass Start Play control																							FPGA	Device Driver &	
																			,						PEU and	

154	150	148		Ħ	
"WODM_OPEN" message with	IVR App invokes system function call to open prerecorded announcement file			IVR APP	
		PBX invokes TSP/TAPI function call to inform IVR App. that WAV and Actual Port Objects have been coupled together		PBX	
				VSR (DAL)	that the transfer that the tra
				PEU Objec	
				WAV Port	To the state of th
				Actual Port	
				WAV Driver	
				Device Driver & FPGA	
			Any play data packets found with this stream ID are sent to Port DSP to mix with other signals such as TDM data and tone data.	PEU and DSP	PEU start monitoring "play data" packet identified by Stream ID for the Port

Device Lot open. WAV channel for play data. 156 Variable of the play data Play Play	Return OK
WAV channel for play data. S S WAV channel for play data. S VSR looks Device ID to play data. VSR looks Device ID in a table and returns handle for that WAV port. PBX VSR Device ID in a table and returns handle for that WAV port. Return OK if not	
WAV channel for play data S IVR APP PBX	
play data. S S S S S S S S S S S S S	
WAV channel for play data. 5 play data VSR looks Device ID in a table and returns handle for that WAV port. IVR APP PBX VSR PEU WAV Port	Port
WAV channel for play data.	Actual
WAV channel for play data.	
WAV channel for play data	
WAV channel for play data.	
WAV channel for play data	
WAV channel for play data.	
WAV channel for play data	
WAV channel for play data.	
WAV channel for play data	
WAV channel for play data	
WAV channel for play data.	
WAV channel for play data	
nnel for	
DATION ON O	

166		ID IVR PBX VSR APP (DAL)
		X VSR (DAL)
		VSR PEU (DAL) Object
WAV Port 208 byte play data packet from the buffer and sets PEU and Port		WAV Port
		Actual Port
	WAV Driver passes buffer to WAV Port designated by Device ID.	Actual WAV Driver Port
	·	Device Driver & FPGA PEU and DSP
		PEU and DSP

170 to 184	168	
		address to all? (broadcast) and Stream ID. Write play data packet to Play Data Queue in shared memory. If given Play Data buffer is not empty, look for Play Data Queue then write the next packet upon queue availability.
		ess to Strea Strea pack pack ie in ie in in ie in in ie in ie in ie in ie in ie in ie in ie in ie in ie in ie in ie in ie in ie in i
		m ID et to shard lay I loc /, loc /, loc ue the ue the ty.
		"(b]). Wr Play Play and maked m Data Data A for hen w bon q
		oadcoatcoatcoatcoatcoatcoatcoatcoatcoatcoat
		ast) lay lay y y y the
		then the second that the secon
		11 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
		<u></u>
,	FPC ever Driv fron Que for c then data	
	iA go y 26 ver to re the ue in each pass pacl	
	mse mse get Play NAN WAN ket to kets 1	
	FPGA generates inter every 26 msec for De Driver to get one pacl from the Play Data Queue in shared mem for each WAV channethen passes each play data packet to FPGA. FPGA delivers the pladata packets to all PE	
	FPGA generates interrupt every 26 msec for Device Driver to get one packet from the Play Data Queue in shared memory for each WAV channel then passes each play data packet to FPGA. FPGA delivers the play data packets to all PEUs.	
	upt ice et et	
Interface logic of stores e broadca specific address packet puts in receive and into MC. No reads explay da packet compar address stream		
FPGA Interface logic of PEU stores each broadcast or specifically addressed packet and puts in receive FIFO and interrupts MC. MC reads each play data packet and compares address and stream ID to		
Interface logic of PEU stores each broadcast or specifically addressed packet and puts in receive FIFO and interrupts MC. MC reads each play data packet and compares address and stream ID to		
<u>"</u>		

			-	
ordered be				
PBX has				
tone data				
writes any				_
port. MC				
mapped to				
buffer of DSP			• • • • • • • • • • • • • • • • • • • •	
TDM data				
moved to				
bus and				
from TDM				
port taken				
data for each				
msec. TDM				
every 26				
that occurs				
from the DSP				
an interrupt				
in response to				
data directed				
which play				
for port to				
buffer of DSP				<u>-</u> -
play data				
moved to				
stream Ids				
matching				
packets with				
Payloads of				
Objects.			_	
Actual Port	, ,			
received from				-
messages				
control				_
stream ID				
built from				
routing table	e from that not it is soull that these	Marif than H" H" H that was		
	17 11 11 11 11 11 11 11 11 11 11 11 11 1	The state of the s		

				-
Data				
directed.				
data is				
port to which				_
assigned to		-1		
bus timeslots				
local TDM				
data onto				
and drives				
sized chunks				
into timeslot				-
TDM data				
outbound				
breaks				
port. DSP				
buffer for that				
TDM data				
into outbound				
for each port				
weighted data				
resulting				
writes				
e fashion and				
programmabl				
source in				
from each				
weight data				
mixing to				
control			_	
does volume	•			_
each PEU				
port. DSP on				
DSP for that				
data buffer of				
port into tone				-
played on			-	

					II W Chaliff model	Արդի (կոս)։ ԱՐ ԱՐ ԱՐ ԱՐ Արդի առաժ Կերտ Կոսի առոժ ԱՐ հրատան Արդի ուսում։ Արդի ուսում։			analog signals at CODEC of each port.
D	IVR APP	РВХ	VSR (DAL)	PEU Object	WAV Port	Actual Port	WAV Driver	Device Driver & FPGA	PEU and DSP
186				:					User presses DTMF tone that indicate user wants to leave a voice mail.
188									DTMF tones are detected by
									MC generates a "DTMF detected" packet with value then send it to FPGA
					-			Pass it to VSR	
			Dispatch it to the						
			Actual Port Object						
			mapped to						
			which the						
			tones were received.			-	,		
						Send "DTMF Detected"			
						message to PBX			
		PBX invokes							

indicating DTMF detected with	value

												Ш
					_	<u> </u>				 		
					operation.	"WODM_RESET" message with Device ID to stop Playing	Announcement" file. IVR App generates	close opened "Prerecorded	IVR App terminates Playing prerecorded announcement.	IVR App recognizes DTMF value is to leave a voice mail.		IVR APP
												РВХ
											(DAL)	VSR
											ect	PEU
		Data from the buffer. Invoke "Stop Playing" Actual Port Object function call	Stop generating packets for Play									WAV Port
	Actual Port Object generates "Stop Playing" control packet and write it to shared memory											Actual Port
				Invoke "Stop Play" WAV Port Interface function call.								WAV Driver
Pass it to PEU						,		_			Driver & FPGA	Device
				,)			and DSP	PEU

Ħ

IVR APP

PBX VSR

PEU

WAV Port

Actual

WAV Driver

Device

PEU and

	(DAL)	(DAL) Object		Port		Driver	DSP
						æ	
						FPGA	
							PEU Stop
							playing data
							packet by
							removing
							Stream ID
						-	from the table
							for addressed
							port
			Return all				
			buffers waiting				
			to play.				
					Send message to IVR		
	<u>.</u>	· · · · ·			App indicating buffer is		
					done. Repeat for all		
					returned buffers.		
IVR App generates							
"WODM_UNPREPARE"							
message with Device ID							
for all prepared buffers.							
					Either OS or WAV		
					Driver deletes provided		
			:		buffer space.		

	,				190					Ш
				IVR App generates "WIDM_OPEN" message with Device ID to open record channel.	IVR App invokes system function call to create/open a file to store recorded data.				IVR App generates "WODM_CLOSE" message with Device ID.	IVR APP
										РВХ
	VSR looks up Device ID in a table and returns handle.									VSR (DAL)
										PEU Objec t
							Set flag to indicate it is closed.			WAV Port
										Actual Port
		Invoke VSR Interface method to obtain handle of WAV Port with specified Device ID	WAV driver inquires of OS where VSR API is. OS responds with a pointer thereto			WAV Driver releases VSR		Invoke "Close Play" WAV Port Interface function.		WAV Driver
	•									Device Driver & FPGA
22										PEU and DSP

			192								
IVR APP			IVR App generates "WIDM_PREPARE" message	with Device ID and size to	provide buffer space in WAV Driver.			IVR App generates "WODM_PREPARE" message	with Device ID and size to	provide buffer space in WAV Driver – second buffer.	
PBX											
(DAL)				•							
Objec t											
WAV Port Return OK if	Return OK if not opened yet										
Actual Port			:		12						
WAV Driver WAV driver uses handle to invoke "Open Record" WAV Port Interface function.		Return OK				Either OS or WAV Driver	App. and return buffer address.				Either OS or WAV Driver provides buffer space for the App. and return buffer
Device Driver & FPGA											
PEU and DSP											

	192				Ш
	194 IVR App. invokes system function call to read data				IVR APP
					PBX
				(DAL)	PBX VSR
			•	(DAL) Objec	PEU
					PEU WAV Port
					Actual Port
					WAV Driver
		FPGA	%	Driver	Device
23			DSP	and	PEU

204	202	200	198		
	·		IVR App generates "WIDM_START" message with Device ID.		from the opened file into the buffer. Then App generates "WIDM_ADDBUFFER" message with Device ID and address of buffer. App repeats if buffer is available.
	WAV Port invokes "Record Start" function of Actual Port Object registered as a listener in this WAV channel.			Queue buffer.	
Actual Port Object generates "Start Sending Record Data" control packet and	d d				
		WAV Driver invokes "Resume Recording" WAV Port Interface function.		WAV Driver invokes "Add Buffer" WAV Port Interface function to pass buffer to save recorded data.	

				1	H thank and that to	shared	shared memory		
	IVR	РВХ	VSR (DAL)	PEU	WAV Port	Actual Port	WAV	Device Driver &	PEU and DSP
	APP		`	Object			Driver	FPGA	
208								Device Driver	
								pass it to FPGA	
							-	and FPGA deliver	
				_				it to the addressed	
								PEU through	
					•			Packet Switched	
								Bus or other	
								control channel	
210									PEU FPGA
									BUS interface
									logic takes
									control packet
									off packet bus
									and passes it to
212									MC reads
	_							•	packet and sets
									state indicating
					****				recording of
									data from the
									designated port
									has started.
214									DSP interrupts
1					•	-1			MC every 26
220									msec. When
									start recording
							-		flag is set, MC
									responds by
									retrieving a 208
									byte payload
	•							·	section of a
		-							record data
				_	,				packet from

222	
,	
	"
rez 26 dai paa dis	
Device Driver reads FIFO every 26 msec and retrieves record data packet and passes it to VSR dispatcher object	
every ld scord t and VSR object	
	DSP buffer file each port that recording. No packetizes date into record date info indicating heade info indicating headed to FPGA white and transmit over packet switched bus when has a transmit toke or is polled file transmit to assigned to record data for this port. Switch card FPGA receive packet and writes it into FIFO.
	DSP buffer for each port that is recording. MC packetizes data into record data packet by adding header info indicating source and type field indicating it is record data Packet passed to FPGA which adds CRC bits and transmit over packet switched bus when has a transmit token or is polled for transmission or using timeslot assigned to record data for this port. Switch card FPGA receives packet and writes it into FIFO.

224	
by	
VSR dispatches message to Actual Port Object that is addressed by source address.	
	1 1
	# # # # # # # # # # # # # # # # # # #
	the feet of the first that the state of the second that the second that the second that the second the second the second that
	in response to polls asking for new messages

234	232	230	228	226	Œ
IVR App invokes system call					IVR APP
					РВХ
					VSR (DAL)
					PEU Object
		If buffer is full, return buffer to WAV Driver.	WAV Port copies it to record data buffer.		WAV Port
				Pass address of payload portion of Record Data packet to WAV Port that is registered as listener.	Actual Port
	WAV Driver sends message using IVR App callback address specified in the Opening process informing it that Recording Buffer is done.				WAV Driver
			-		Device Driver & FPGA
					PEU and DSP

							·	1	
							D		
IV/D A man desides to							IVR APP		to write data in the buffer to opened file.
<u></u>	r c:			··•.			P		
	PBX invokes TAPI function call indicating "OnHook"						PBX		
			Actual Port Object	dispatches			VSR (DAL)		
•							PEU Objec t		
							WAV Port		
		Send "OnHook" message to PBX					Actual Port		F
			_				WAV Driver		
					Pass it to VSR		Device Driver & FPGA		
						User hangup handset. MC Detects Status, generates "OnHook detected" control packet with PEU and Port info then sends packet to device driver	PEU and DSP		

					IVR APP	stop recording. Invoke system function call to close opened file on the hard disk. IVR App generates "WIDM_RESET" message with Device ID to stop Recording operation.
					РВХ	
					VSR (DAL)	
					PEU Object	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
			Invoke "Stop Recording" Actual Port Object function call of Actual Port Object in WAV channel.		WAV Port	
		Actual Port Object generates "Stop Recording" control packet and write it to shared memory			Actual Port	
				Invoke "Stop Record" WAV Port Interface function call.	WAV Driver	
	Pass it to PEU			•	Device Driver & FPGA	
PEU Stop sending record data				i i	PEU and DSP	

				territ for the state that there there were the state that the state the stat	11.111			packet
			1	Returns buffers waiting for recording.				
						Send message to IVR App indicating buffer is done. Repeat for all returned buffers.		
IVR APP	РВХ	VSR (DAL)	PEU Object	WAV Port	Actual Port	WAV Driver	Device Driver	PEU and DSP
IVR App generates "WIDM_UNPREPA RE" message with Device ID for all prepared buffers.								
						Either OS or WAV Driver deletes provided buffer space.		_
IVR App generates "WIDM_CLOSE" message with Device ID				,				
						Invoke "CloseRecord" WAV Port Interface function.		
ì				Set flag to indicate it is closed.				
						WAV Driver releases VSR.		
IVR App saves								

	for recorded voice mail	recorded file into area
		ion and the transfer of the state of the sta
-		

•

.